

AMENDMENTS TO THE SPECIFICATION

In paragraph [0038] please amend as reflected in the following marked-up version of the paragraph:

[0038] Assuming a visual software service composition and assembly tool, such as HyperService™ Studio with a snapshot of a definition in FIGURE 2, and an automatic flow platform, such as the HyperService™ platform, ~~available at www.nextaxiom.com, we~~. We will now describe how such a composition tool and platform can be extended to accommodate declarative semantic-based recovery and transactional support for nested composite web services.

In paragraph [0049] please amend as reflected in the following marked-up version of the paragraph:

[0049] To provide the automatic runtime behavior for transactional support, under the declarative and semantic-based visual techniques discussed in the present invention, the automation platform starts the invocation of a composite service instance, declared as transactional, by creating a persistent instance of an Invocation Map based on the composite service definition. Then, once services within each HyperCycle are invoked, if all the services are consumed successfully (or, even if some services failed, by the had a defined routing upon failure that was successful) and if the containing composite service is a root service (i.e. one that is contained within another composite service), the nested contained composite services and all their nested composite services to the depth of nesting, within the containing composite service, are committed together with the entire unique context associated to each library, if any, and the context associated to the shared memory. If all the services contained within a composite service execute successfully (or, even if some services failed, by the had a defined routing upon failure that was successful), but the containing service is not a root containing service, the contained composite service will wait until it receives a commit or rollback call from the context of the containing service. Furthermore, if any of the services within a HyperCycle fail ~~fail~~, without a predefined routing upon failure, all the context within the Invocation Map of the composite

service instance will be rolled back. This rollback will result in nested rollback, if any of the contained services are composite services, and it includes the rollback of shared memory and all the associated atomic libraries as discussed earlier. Once an invocation Map enters a rollback or a commit state, the connectivity of the rollback/compensation/commit ports of the contained services are implemented as follows: 1) If a flow port marked for rollback is connected, and if the port is declared as an overwriting port, the built-in rollback of the underlying service instance will not be requested by the automation platform; instead, the services connected to that port are invoked as the rollback act; 2) however, if the port is declared as an extending port, then after the completion of the rollback of the underlying service, the services connected to the port are invoked to extend the act of rollback. Upon rollback, if the compensation port of a service is connected to other services, those services are invoked by the automation platform. The act of rollback, is performed in a natural back tracking order with respect to all the contents of a composite service.